一、名词解释（每小题 2 分）

1. **地址映射**：将程序中的逻辑地址转换为内存中的物理地址的过程，是操作系统实现内存管理的核心功能之一，确保程序能正确访问内存单元。
2. **动态重定位**：在程序执行过程中，每次访问内存时才进行地址转换的方式，通常通过硬件地址转换机构（如重定位寄存器）实现，允许程序在内存中移动。
3. **虚拟存储器**：一种基于内存和外存的存储管理技术，通过将部分程序数据换入换出，使程序能以逻辑地址空间运行，突破物理内存容量限制，为用户提供远超实际内存的“虚拟”存储空间。
4. **静态链接**：在程序装入内存前，由链接程序将多个目标模块及所需库函数链接成一个完整的可执行文件，后续不再修改，链接时确定所有符号地址。
5. **对换**：操作系统将内存中暂时不运行的进程或部分数据换出到外存，释放内存空间供其他进程使用，待需要时再将其换入内存，以提高内存利用率。
6. **设备驱动程序**：操作系统中负责控制特定 I/O 设备的程序，它接收上层软件的请求，将其转换为设备能理解的指令，实现设备的具体操作（如读写、启停等）。
7. **SPOOLing**：即假脱机技术，通过在磁盘上开辟输入井和输出井，将独占设备虚拟为共享设备。用户程序的输入输出请求先提交到井中，再由系统统一调度设备处理，提高设备利用率。
8. **I/O 通道**：一种专门负责输入输出的硬件，它能独立于 CPU 执行通道程序，管理多个设备的 I/O 操作，减轻 CPU 负担，提高系统效率。
9. **文件系统**：操作系统中管理文件的机构，它负责文件的创建、删除、读写、保护等，并提供文件的组织、存储和检索功能，方便用户使用外存。
10. **目标文件**：源程序经编译后生成的二进制文件，它包含程序的机器指令和数据，但尚未完成链接，不能直接运行。
11. **文件的逻辑结构**：从用户角度看到的文件组织形式，即文件中数据的排列方式和逻辑关系，如流式文件、记录式文件等，方便用户存取和理解。
12. **有结构文件**：又称记录式文件，文件由若干个逻辑记录组成，每个记录包含相关的数据项，记录之间有明确的逻辑关系（如顺序、索引等）。
13. **位示图**：一种用于管理外存空闲空间的数据结构，它用一个二进制位表示一个磁盘块的状态（0 表示空闲，1 表示已分配），通过位运算可快速查找和分配空闲块。
14. **程序接口**：操作系统为应用程序提供的接口，用户程序通过调用这些接口（如

系统调用）来请求操作系统服务（如文件操作、进程控制等）。

15. **系统调用**：应用程序向操作系统请求服务的一种方式，它通过特定的指令（如陷阱指令）使程序从用户态切换到核心态，由操作系统执行相应功能后返回结果。
16. **I/O 中断**：当 I/O 设备完成操作或出现错误时，向 CPU 发送的中断信号，CPU 响应后暂停当前任务，转去执行中断处理程序（如处理数据、报告错误等）。
17. **文件管理系统**：即文件系统，同第 9 题。
18. **文件**：由创建者定义的一组相关信息的集合，存储在外部存储介质上，具有文件名，是操作系统管理信息的基本单位。
19. **文件的逻辑结构**：同第 11 题。
20. **文件的物理结构**：文件在外部存储介质上的实际存储方式，即数据在磁盘上的排列顺序和组织形式，如连续结构、链接结构、索引结构等。

二、填空题（每空 1 分）

1. 地址映射
2. 绝对装入、可重定位装入、动态运行时装入
3. 静态链接、装入时动态链接、运行时动态链接
4. 空闲分区表、空闲分区链
5. 内存紧凑（或拼接）
6. 固定、系统、可变、用户程序（或进程）
7. 虚拟性、离散性
8. 文件管理程序、被管理的文件、文件目录
9. 分配算法、分配与回收操作、内存保护
10. 碎片（或外部碎片）
11. 分页存储管理、分段存储管理、段页式存储管理
12. 快表（或转换检测缓冲区 TLB）
13. 逻辑地址、物理地址
14. 满足程序模块化需求、便于实现共享、便于实现保护、便于动态链接、能更好地适应程序的动态增长

15. 地址变换机构

16. 局部性、时间局部性、空间局部性

17. 处理器、通道能独立执行通道程序、通道专门用于 I/O 操作

18. 顺序文件、索引文件、索引顺序文件

19. 单级索引、多级索引、混合索引

20. 动态重定位

21. 用户层 I/O 软件、设备独立性软件、设备驱动程序

22. 连续结构、链接结构、索引结构、索引结构

23. 高速缓存（Cache）、主存储器（内存）、辅助存储器（外存）

24. 连续分配、离散分配

25. 存储介质特性、文件的存取方式

26. 无结构文件（流式文件）、有结构文件（记录式文件）、数据库文件

27. 连续分配、链接分配、索引分配

28. 绝对路径、相对路径

29. 空闲盘块链、空闲盘区链

30. 静态共享、动态共享

31. 块设备、字符设备、独占设备、共享设备、虚拟设备

32. DMA 方式、通道方式

33. 单缓冲、循环缓冲、缓冲池

34. 寻道时间、传输时间

三、简答题（每小题 6 分）

1. **计算机存储器系统主要有哪些层次？各个层次又包含哪些内容？试从存储性能角度对存储系统进行分析。**

层次及内容：存储器系统从快到慢、从贵到廉分为三层：

高速缓存（Cache）：位于 CPU 内部或附近，速度最快、容量最小、价格最高，用于存储 CPU 近期频繁访问的数据和指令。

主存储器（内存）：速度次之、容量中等、价格适中，用于存放当前运行的程序和数据，CPU 可直接访问。

辅助存储器（外存）：如磁盘、光盘等，速度最慢、容量最大、价格最低，用于长期存储程序和数据，CPU 不能直接访问，需先调入内存。

性能分析：通过“Cache - 内存 - 外存”的层次结构，结合程序访问的局部性原理，使存储系统在速度上接近 Cache，在容量和价格上接近外存，平衡了速度、容量和成本的矛盾，提高了系统整体性能。

2. **简述计算机程序从代码编写到运行完毕所经历的主要过程。**

编写阶段：用户用高级语言（如 C、Java）或汇编语言编写源程序。

编译阶段：编译器将源程序转换为目标文件（二进制代码，含指令和数据，但未链接）。

链接阶段：链接程序将多个目标文件及所需库函数链接成一个完整的可执行文件，确定符号地址。

装入阶段：操作系统将可执行文件装入内存，完成地址映射（逻辑地址→物理地址）。

运行阶段：CPU 执行内存中的程序指令，过程中可能发生中断、I/O 操作等，直至程序执行完毕，释放资源。

3. **程序链接的主要方式有哪些？它们的异同点是什么？**

主要方式：静态链接、装入时动态链接、运行时动态链接。

相同点：均需将多个目标模块及库函数组合，确定符号地址，形成可执行程序。

不同点：

静态链接：在程序装入前完成链接，生成单一可执行文件，运行时无需重新链接，但文件体积大，更新库函数需重新编译链接。

装入时动态链接：在程序装入内存时链接，仅链接所需模块节省内存，但装入速度较慢。

运行时动态链接：在程序执行过程中，当需要某模块时才链接，进一步节省内存，便于模块共享和更新，但可能增加运行时开销。

4. **内存动态分区分配算法根据搜索空闲区的方式可分为哪些类型，它们又分别有哪些典型的分配算法？请逐一简述这些算法的核心思想。**

类型及算法：

首次适应算法：从空闲分区表 / 链的起始位置开始搜索，找到第一个能满足需求的空闲分区，将其分配。优点是简单，缺点是低地址区易产生碎片。

循环首次适应算法：从上次分配的分区后开始搜索，找到第一个合适的空闲分区。可使空闲分区分布更均匀，减少碎片集中。

最佳适应算法：搜索所有空闲分区，选择最小的、能满足需求的分区。可减少浪费，但会产生更多小碎片，搜索开销大。

最坏适应算法：选择最大的空闲分区分配。可减少小碎片，但可能破坏大分区，不利于大程序分配。

5. **列举典型的内存动态分区算法和 CPU 调度算法，并分析两者之间的异同点。**

内存动态分区算法：首次适应、循环首次适应、最佳适应、最坏适应等。

CPU 调度算法：先来先服务、短作业优先、优先级调度、时间片轮转等。

相同点：均需根据一定策略进行资源（内存 / CPU）分配，目标是提高资源利用率和系统效率。

不同点：

管理对象不同：内存算法管理内存空间，CPU 调度算法管理 CPU 时间。

目的不同：内存算法旨在合理分配内存，减少碎片；CPU 调度算法旨在提高 CPU 利用率，缩短作业周转时间等。

触发时机不同：内存分配在进程创建或请求内存时触发；CPU 调度在进程状态变化（如运行→阻塞）时触发。

6. **列举典型的页面置换算法和 CPU 调度算法，并分析两者之间的异同点。**

页面置换算法：最佳置换（OPT）、先进先出（FIFO）、最近最久未使（LRU）、时钟置换（CLOCK）等。

CPU 调度算法：同第 5 题。

相同点：均为资源分配策略，通过合理选择对象（页面 / 进程）提高系统性能。

不同点：

应用场景不同：页面置换用于虚拟存储中，当内存不足时换出页面；CPU 调度用于选择进程占用 CPU 运行。

优化目标不同：页面置换旨在减少缺页率；CPU 调度旨在优化周转时间、响应时间等。

依据不同：页面置换基于页面访问历史（如 LRU）；CPU 调度基于进程属性（如优先级、运行时间）。

7. **列举典型的页面置换算法和磁盘调度算法，并分析两者之间的异同点。**

页面置换算法：同第 6 题。

磁盘调度算法：先来先服务（FCFS）、最短寻道时间优先（SSTF）、扫描（SCAN）、循环扫描（CSCAN）等。

相同点：均需选择“对象”（页面 / 磁道请求）进行处理，以提高系统效率，减少操作开销。

不同点：

- 处理对象不同：页面置换针对内存页面；磁盘调度针对磁盘 I/O 请求的磁道顺序。
- 优化目标不同：页面置换减少缺页带来的磁盘 I/O 开销；磁盘调度减少寻道时间，提高磁盘访问速度。
- 依赖因素不同：页面置换依赖页面访问局部性；磁盘调度依赖磁道位置分布。

8. **分别简述分页存储、分段存储和段页式存储的地址变换过程。**

分页存储：

- 将逻辑地址拆分为页号和页内偏移量。
- 查页表，得到页号对应的物理块号。
- 物理地址 = 物理块号 × 块长 + 页内偏移量。（可通过快表加速查找）

分段存储：

- 将逻辑地址拆分为段号和段内偏移量。
- 查段表，得到段的基址和段长，检查偏移量是否越界。
- 物理地址 = 段基址 + 段内偏移量。

段页式存储：

将逻辑地址拆分为段号、段内页号、页内偏移量。

查段表得到该段的页表基址。

查页表（根据段内页号）得到物理块号。

物理地址 = 物理块号 × 块长 + 页内偏移量。（需两次查表，可通过快表优化）

9. **简述中断处理程序的各个处理步骤。**

1.中断请求：I/O 设备或其他事件（如时钟、故障）向 CPU 发出中断信号。

2.中断响应：CPU 在执行完当前指令后，检测到中断请求，保存当前程序状态字（PSW）和断点地址，关闭中断。

3.中断判优：确定中断源的优先级，选择最高优先级的中断处理。

4.中断服务：执行相应的中断处理程序（如读取设备数据、处理错误等）。

5.中断返回：恢复被中断程序的 PSW 和断点地址，开启中断，继续执行原程序。

10. **从用途、数据类型、组织和管理方式等角度简述文件类型分类。**

按用途：系统文件（操作系统程序）、用户文件（用户创建的文件）、库文件（标准函数库）。

按数据类型：文本文件（字符序列）、二进制文件（非字符数据，如可执行文件、图像）。

按组织方式：无结构文件（流式文件，无记录划分）、有结构文件（记录式文件，含多个逻辑记录）。

按管理方式：普通文件（用户数据和程序）、目录文件（管理文件的目录项）、特殊文件（I/O 设备的抽象表示）。

11. **请详述文件目录的分类及相应查询方式。**

分类：

单级目录：整个系统一个目录，所有文件在同一目录下。

两级目录：分为主目录（存放用户目录项）和用户目录（存放该用户的文件目录项）。

树型目录：目录呈树状结构，有一个根目录，各级子目录可包含文件和下一级目录。

查询方式：

单级目录：直接按文件名线性查找。

两级目录：先查主目录找到用户目录，再在用户目录中查文件名。

树型目录：通过路径名查询，包括绝对路径（从根目录开始）和相对路径（从当前目录开始），按路径逐层查找目录项。

12. **请简述文件结构的三种主要组织方式，并对比分析各自优劣。**

连续结构：文件数据在磁盘上连续存储。

优点：读写速度快（可顺序访问），管理简单。

缺点：创建文件时需预分配空间，易产生外部碎片，文件不易扩展。

链接结构：文件数据分散存储，通过指针链接各块。

优点：空间利用率高，文件易扩展。

缺点：随机访问慢（需沿指针查找），指针占用存储空间，可靠性差（指针损坏导致文件丢失）。

索引结构：为文件建立索引表，记录各块地址。

优点：随机访问快，文件易扩展，可靠性高。

缺点：索引表占用空间，小文件的索引表开销相对较大。

13. **请分别简述内存和外存的存储分配空间分配方式，并对比分析它们之间的异同点。**

内存分配方式：

连续分配：单一连续区、分区管理（固定 / 动态），内存块连续。

离散分配：分页、分段、段页式，内存块不连续。

外存分配方式：连续分配、链接分配（隐式 / 显式）、索引分配，同文件物理结构的分配方式。

相同点：均需管理空闲空间，目的是高效利用存储资源，支持存储对象的创建和释放。

不同点：

速度要求：内存分配需快速响应（CPU 直接访问），外存分配对速度要求较低。

碎片影响：内存碎片影响大（内存资源紧张），外存碎片影响较小（容量大）。

分配单位：内存以块 / 页 / 段为单位，外存以盘块为单位，单位大小通常更大。

14. **请分别简述提高磁盘 I/O 速度的多种途径。**

采用高效的磁盘调度算法（如 SSTF、SCAN），减少寻道时间。

预读和延迟写：提前读取可能需要的数据，延迟写入磁盘（先存缓存），减少 I/O 次数。

磁盘高速缓存：在内存中开辟缓冲区，暂存磁盘数据，减少实际磁盘访问。

RAID 技术：将多个磁盘组合，通过并行读写、数据冗余等提高速度和可靠性。

优化文件物理结构：采用索引结构或连续结构，减少磁盘寻道和旋转延迟。

合理分配磁盘空间：避免文件过于分散，减少碎片，提高读写效率。

使用更快的接口和设备：如采用 SSD（固态硬盘）替代机械硬盘，或使用 SATA、NVMe 等高速接口。